1. A branch prediction apparatus in a processor including address selection logic for providing a fetch address to an instruction cache, the fetch address used to select lines of the instruction cache, the apparatus comprising:

first and second branch predictors, for providing first and second target address predictions of a branch instruction to the address selection logic;

instruction decode logic, configured to receive and decode said branch instruction and to generate a type thereof; and

branch control logic, configured to control the address selection logic to select said first prediction as the fetch address, said first prediction selecting a first line of the instruction cache;

wherein said branch control logic is further configured to subsequently selectively control the address selection logic, based on said branch instruction type, to select said second prediction as the fetch address, said second

prediction selecting a second line of the instruction cache.

2. The apparatus of claim 1, further comprising:

comparison logic, coupled to said first and second branch predictors, for comparing said first and second target address predictions.

3. The apparatus of claim 2 wherein said type includes a specification of whether said branch instruction is a return type branch instruction.

4. The apparatus of claim 3, wherein said branch control logic controls the address selection logic to select said second target address prediction if said branch instruction type is a return instruction and said first and second predictions miscompare.

5. The apparatus of claim 4, wherein said second branch predictor comprises a call/return stack for providing said second target address prediction of said return instruction.

6. The apparatus of claim 2, wherein said type includes a specification of whether said branch instruction is a program counter-relative type branch instruction.

7. The apparatus of claim 6, wherein said branch control logic controls the address selection logic to select said second target address prediction if said branch instruction type is a program counter-relative branch instruction and said first and second predictions miscompare.

8. The apparatus of claim 7, wherein said second branch predictor comprises an arithmetic unit for calculating said second target address prediction based on an instruction pointer of said branch instruction.

9. The apparatus of claim 8, wherein said arithmetic unit calculates said second target address prediction using said instruction pointer of said branch instruction.

10. The apparatus of claim 2, wherein said type includes a specification of whether said branch instruction is a direct type branch instruction.

11. The apparatus of claim 10, wherein said branch control logic controls the address selection logic to select said second target address prediction if said branch instruction type is a direct branch instruction and said first and second predictions miscompare.

12. The apparatus of claim 2, wherein said type includes a specification of whether said branch instruction is an indirect type branch instruction.

13. The apparatus of claim 12, wherein said branch control logic controls the address selection logic not to select said second target address prediction if said branch instruction type is an indirect branch instruction.

14. The apparatus of claim 13, wherein said second branch predictor comprises a branch target buffer for caching branch target addresses of previously executed indirect branch instructions.

15. The apparatus of claim 2, wherein said type includes a specification of whether said branch instruction is a conditional type branch instruction.

16. The apparatus of claim 15, wherein said branch control logic controls the address selection logic to select said second target address prediction if said branch instruction type is a conditional branch instruction and said first and second predictions miscompare.

17. The apparatus of claim 15, wherein said first and second predictors provide first and second direction

predictions of said conditional branch instruction to said branch control logic for predicting whether said conditional branch instruction will be taken.

18. The apparatus of claim 17, further comprising:

second comparison logic, coupled to said first and second branch predictors, for comparing said first and second direction predictions of said conditional branch instruction.

19. The apparatus of claim 18, wherein said branch control logic controls the address selection logic to select an instruction pointer of a next sequential instruction to said conditional branch instruction as the fetch address if said second direction prediction predicts said conditional branch instruction will not be taken.

20. The apparatus of claim 19, wherein said branch control logic controls the address selection logic to select said next sequential instruction pointer if said second direction prediction predicts said conditional branch instruction will not be taken and said first and second direction predictions miscompare.

21. The apparatus of claim 2, wherein said branch control logic subsequently selectively controls the address selection logic based on said branch instruction type to select said second prediction as the fetch address if said first and second predictions do not match.

22. The apparatus of claim 1, wherein said branch instruction type comprises an x86 branch instruction type.

23. The apparatus of claim 1, wherein said first branch predictor receives the instruction cache fetch address and provides said first target address prediction in response to the fetch address.

24. The apparatus of claim 23, wherein said first branch predictor provides said first target address prediction in response to the fetch address whether or not a branch instruction is present in a third line of the instruction cache, said third instruction cache line selected subsequent to selection of said first instruction cache line.

25. The apparatus of claim 23, wherein said first branch predictor provides said first target address

prediction prior to said instruction decode logic decoding said branch instruction.

26. The apparatus of claim 1, wherein said first branch predictor comprises a branch target address cache indexed by the instruction cache fetch address.

27. The apparatus of claim 1, wherein said first branch predictor comprises a speculative call/return stack.

28. A branch prediction apparatus in a processor, comprising:

first and second branch predictors, for making first and second predictions of a branch instruction;

comparison logic, coupled to said first and second branch predictors, configured to provide a comparison of said first and second predictions;

instruction decode logic, configured to decode said branch instruction and to generate a type thereof; and

control logic, coupled to said instruction decode logic, for causing the processor to branch based on said first prediction;

wherein said control logic selectively overrides, based on said type of said branch instruction and said comparison, said first prediction with said second prediction.

29. The apparatus of claim 28, wherein said instruction decode logic decodes and generates said branch instruction type subsequent to said first branch predictor making said first prediction.

30. The apparatus of claim 28, wherein said second branch predictor makes said second prediction in response to said instruction decode logic decoding said branch instruction.

31. The apparatus of claim 28, wherein said second branch predictor comprises a branch history table for providing a direction prediction in said second prediction, said direction prediction for predicting whether said branch instruction will be taken or not taken.

32. The apparatus of claim 28, wherein if said branch type is a conditional branch type, said comparison logic compares first and second direction predictions in said first and second predictions, respectively.

33. The apparatus of claim 32, wherein if said first and second direction predictions do not match, said control logic overrides said first prediction with said second prediction.

34. The apparatus of claim 28, wherein said comparison logic compares first and second target addresses in said first and second predictions, respectively.

35. The apparatus of claim 34, wherein said control logic selectively overrides, based on said type of said branch instruction, said first prediction with said second prediction by causing the processor to branch to said second target address if said first and second target addresses do not match.

36. A pipelined processor, comprising:

a speculative branch predictor, for making a speculative prediction of a branch instruction;

control logic, coupled to said speculative branch predictor, for branching the processor based on said speculative prediction;

instruction decode logic, configured to decode and generate a type of said branch instruction; and

a non-speculative branch predictor, coupled to said instruction decode logic, for making a non-speculative prediction of said branch instruction;

wherein said control logic subsequently selectively branches the processor based on said non-speculative prediction and said branch instruction type.

37. The processor of claim 36, further comprising:

an instruction cache, coupled to an address bus for receiving a fetch address, said fetch address selecting a line of instructions for provision to said instruction decode logic.

38. The processor of claim 37, wherein said speculative branch predictor makes said speculative prediction even though a possibility exists that no branch instruction is present in said line of instructions.

39. The processor of claim 36, wherein said non-speculative branch predictor makes said non-speculative prediction in response to said instruction decode logic decoding said branch instruction.

40. A pipelined processor, comprising:

a branch target address cache, for providing a speculative target address of an instruction prior to decoding of said instruction;

a target address calculator, for calculating a non-speculative target address of said instruction after said decoding of said instruction; and

a comparator, coupled to said branch target address cache and said target address calculator, for comparing said speculative and non-speculative target addresses;

wherein said processor branches to said speculative target address, wherein the processor subsequently branches to said non-speculative target address if said speculative and non-speculative target addresses miscompare and if said instruction is a type comprised in a first set of instruction types.

41. The processor of claim 40, wherein said first set of instruction types includes a return instruction type.

42. The processor of claim 40, wherein said first set of instruction types includes a program counter-relative branch instruction type.

43. The processor of claim 40, wherein said first set of instruction types includes a conditional branch instruction type.

44. A pipelined processor, comprising:

a branch target address cache, for providing a speculative direction of whether a presumed conditional branch instruction will be taken or not taken, said speculative direction provided prior to decoding of said presumed conditional branch instruction;

a branch history table, for providing a non-speculative direction of said instruction, said non-speculative direction provided after said decoding of said presumed conditional branch instruction; and

a comparator, coupled to said branch target address cache and said branch history table, for comparing said speculative and non-speculative directions;

wherein if said speculative and non-speculative directions miscompare the processor branches to a next sequential instruction pointer after said instruction if said non-speculative direction specifies that the instruction will not be taken.

45. A method for branching in a pipelined processor, comprising:

generating a speculative target address of a branch instruction;

branching the processor to said speculative target address;

decoding said branch instruction after said branching;

generating a non-speculative target address of said branch instruction after said decoding;

determining a branch type of said branch instruction;

determining whether said speculative and non-speculative target addresses match; and

selectively branching, based on said branch type, to said non-speculative target address if said speculative and non-speculative target addresses do not match.

46. The method of claim 45, further comprising:

determining if said branch type is a program counter-relative type branch instruction;

wherein said selectively branching comprises branching to said non-speculative target address if said speculative and non-speculative target addresses do not match and if said branch type is a program counter-relative type branch instruction.

47. The method of claim 45, further comprising:

determining if said branch type is a return type branch instruction;

wherein said selectively branching comprises branching to said non-speculative target address if said speculative and non-speculative target addresses do not match and if said branch type is a return type branch instruction.

48. The method of claim 45, further comprising:

determining if said branch type is a conditional type branch instruction;

wherein said selectively branching comprises branching to said non-speculative target address if said speculative and non-speculative target addresses do not match and if said branch type is a conditional type branch instruction.

49. The method of claim 45, further comprising:

determining if said branch type is an indirect type branch instruction;

wherein said selectively branching comprises not branching to said non-speculative target address if said branch type is an indirect type branch instruction.

50. The method of claim 45, further comprising:

generating a non-speculative direction prediction of said branch instruction after said decoding said branch instruction; and

branching to a next sequential instruction pointer after said branch instruction if said non-speculative direction prediction indicates said branch instruction will not be taken.